What is claimed is:

1. A method for modeling a web server, comprising:

identifying a plurality of sub-systems for the server;

representing each sub-system as a queue, with each queue operably coupled together; and

iteratively adjusting an arrival rate and a service time for each queue to account for performance by other queues.

2. The method of claim 1, wherein said plurality of sub-systems comprises one or more of a set comprising a transaction control protocol/internet protocol sub-system, a hypertext transfer protocol sub-system, an input/output sub-system, and an active script component sub-system.

3. The method of claim 1, wherein each sub-system is modeled as a finite-buffer, finite server queueing system.

4. The method of claim 2, wherein said transaction control protocol/internet protocol sub-system comprises a first finite listen queue served by a listener daemon.

5. The method of claim 2, wherein said hypertext transfer protocol sub-system comprises a second finite listen queue served by one or more multi-threaded hypertext transfer protocol daemons with N_{http} separate server threads.

6. The method of claim 2, wherein said input/output sub-system comprises a finite number N_{buf} of network buffers served by an input/output controller.

7. The method of claim 6, wherein said input/output controller serves each network buffer using a polling system.

8. The method of claim 2, wherein said transaction control protocol/internet protocol sub-system TCP/IP is represented as an $M(\lambda_{file}) / M(\tau_{tcp}) / N_{tcp} / 0$ blocking system.

9. The method of claim 2, wherein said hypertext transfer protocol sub-system is represented as an $M(\lambda_{http}) / M(\tau_{http}) / N_{http} / Q_{http}$ queueing system.

10. The method of claim 2, wherein said input/output sub-system is represented as an $M(\lambda_{buf}) / M(\tau_{buf}) / N_{buf} / \infty$ queueing system.

11. A method for modeling a web server, comprising:

(a) identifying for the server a transaction control protocol/internet protocol (TCP/IP) sub-system, a hypertext transfer protocol (HTTP) sub-system, and an input/output (I/O) sub-system;

(b) representing each sub-system as a queuing system;

(c) computing an upper bound performance for said I/O sub-system by assuming a first predetermined blocking value for said TCP/IP sub-system and HTTP sub-system;

(d) computing an upper bound performance for said TCP/IP sub-system and HTTP sub-system by assuming a first predetermined I/O sub-system waiting time;

(e) computing a lower bound I/O performance by assuming a second predetermined blocking value for said TCP/IP sub-system and HTTP sub-system;

(f) computing a lower bound performance for said TCP/IP sub-system and HTTP sub-system by assuming a second predetermined I/O sub-system waiting time; and

(g) repeating steps (c) - (f) to generate successively tighter bounds until convergence.

12. A machine-readable medium whose contents cause a computer system to model a web server, by performing the steps of:

identifying a plurality of sub-systems for the server;

representing each sub-system as a queue, with each queue operably coupled together; and

iteratively adjusting an arrival rate and a service time for each queue to account for performance by other queues.

13. The machine-readable medium of claim 12, wherein said plurality of sub-systems comprises one or more of a set comprising a transaction control protocol/internet protocol sub-system, a hypertext transfer protocol sub-system, an input/output sub-system, and an active script component sub-system.

14. The machine-readable medium of claim 12, wherein each sub-system is modeled as a finite-buffer, finite server queueing system.

15. The machine-readable medium of claim 13, wherein said transaction control protocol/internet protocol sub-system comprises a first finite listen queue served by a listener daemon.

16. The machine-readable medium of claim 13, wherein said hypertext transfer protocol sub-system comprises a second finite listen queue served by one or more multi-threaded hypertext transfer protocol daemons with N_{http} separate server threads.

17. The machine-readable medium of claim 13, wherein said input/output sub-

system comprises a finite number N_{buf} of network buffers served by an input/output controller.

18. The machine-readable medium of claim 17, wherein said input/output controller serves each network buffer using a polling system.

19. The machine-readable medium of claim 13, wherein said transaction control protocol/internet protocol sub-system TCP/IP is represented as an $M(\lambda_{file}) / M(\tau_{tcp}) / N_{tcp} / 0$ blocking system.

20. The machine-readable medium of claim 13, wherein said hypertext transfer protocol sub-system is represented as an $M(\lambda_{http}) / M(\tau_{http}) / N_{http} / Q_{http}$ queueing system.

21. The machine-readable medium of claim 13, wherein said input/output sub-system is represented as an $M(\lambda_{buf}) / M(\tau_{buf}) / N_{buf} / \infty$ queueing system.

22. A machine-readable medium for modeling a web server, comprising:

(a) identifying for the server a transaction control protocol/internet protocol (TCP/IP) sub-system, a hypertext transfer protocol (HTTP) sub-system, and an input/output (I/O) sub-system;

(b) representing each sub-system as a queuing system;

(c) computing an upper bound performance for said I/O sub-system by assuming a first predetermined blocking value for said TCP/IP sub-system and HTTP sub-system;

(d) computing an upper bound performance for said TCP/IP sub-system and HTTP sub-system by assuming a first predetermined I/O sub-system waiting time;

(e) computing a lower bound I/O performance by assuming a second predetermined blocking value for said TCP/IP sub-system and HTTP sub-system;

(f) computing a lower bound performance for said TCP/IP sub-system and HTTP sub-system by assuming a second predetermined I/O sub-system waiting time; and

(g) repeating steps (c) - (f) to generate successively tighter bounds until convergence.